Jet background subtraction

Dennis V. Perepelitsa
University of Colorado Boulder

30 May 2017
sPHENIX Simulations Meeting

Overview

- First (re-)implementation of iterative “ATLAS-style” UE background determination & subtraction for jet reco in a heavy ion environment
    - ➡ capability described in our methods PRC paper which was lost in the sPHENIX/PHENIX software “divorce”, now re-implemented from scratch
    - ➡ mostly (but not fully) implemented the procedure as described
        - ➡ adding core components to *offline/packages/jetbackground*
- UE energy density ρ (or UE energy per tower) estimated for each of three detector layers (CEMC, IHCal, OHCal) and in $\Delta\eta$=0.1 strips
    - ➡ thus, CEMC first cast into a $\Delta\eta \mathrm{x} \Delta\phi$ = 0.1x0.1 tower grid to match I/OHCal
    - ➡ exclude towers within ΔR<0.4 of a “seed” jet from background determination
    - ➡ update tower kinematics, $E^{\mathrm{sub}} = E^{\mathrm{raw}} - \rho(\eta, \mathrm{layer})\, A^{\mathrm{tower}}$, run jet reco again
- Two missing pieces of functionality:
    - ➡ right now, only one iteration (for exclusion regions, using p_{T} > 25 GeV R=0.2 jets run without any subtraction as seeds)
    - ➡ no accounting for possible v_2 modulation (v_2 = 0, Ψ_2 = DNE)

New data nodes / modules

- **ReclusterCEMC** : make a 0.1x0.1-towerized version of the CEMC (e.g. matching the IHCal/OHCal geometry)
  - ➡ writes TOWER_CALIB_CEMC_RETOWER tower container to node
  - ➡ otherwise, jet finding with ~25k EMCal tower inputs is very slow…
- **DetermineTowerBackground** : given unsubtracted CEMC (0.1x0.1-retowerized), IHCal and OHCal towers, determines UE energy per tower (vs. η and layer)
  - ➡ writes these parameters to the node in a **TowerBackground** object
  - ➡ also stores layer-dependent v_2, Ψ_2 information, but unused right now
- **SubtractTowers** : takes the TowerBackground and unsubtracted towers, creates a new set of subtracted towers
  - ➡ additionally, fills in any towers that were not created because they had zero energy before subtraction
  - ➡ writes TOWER_CALIB_CEMC_RETOWER_SUB1, TOWER_CALIB_HCALIN_SUB1, TOWER_CALIB_HCALOUT_SUB1 tower containers to node

Mods to jet finding code

- Major goal: HI jet reconstruction on subtracted towers can be configured just within a standard JetReco module (see example on the next page)
    - ➡ e.g. outputs are sPHENIX Jet objects just in a different JetContainer
- In g4jets, modified Jet.h and TowerJetInput.C to take a new tower container source:
    - ➡ e.g. SRC enum Jet::HCALIN_TOWER_SUB1 for TOWER_CALIB_HCALIN_SUB1 container, etc.
- Created **FastJetRecoSub**, an analogue of FastJetReco but which correctly deals with negative-*E* towers:
    - ➡ sets these towers to have *E* = 1 MeV and their correct original η/ϕ direction for the purpose of jet finding
    - ➡ then, after jet finding, updates the jet kinematics to the four-vector sum of the constituents
        - ➡ this is a key capability needed for any HI jet reco, also described in the methods PRC paper

Example macro

```
RetowerCEMC *rcemc = new RetowerCEMC();
se->registerSubsystem( rcemc );

DetermineTowerBackground *dtb = new DetermineTowerBackground();
dtb->SetBackgroundOutputName("TowerBackground_Sub1");
dtb->SetSeedType( 1 );
se->registerSubsystem( dtb );

SubtractTowers *st = new SubtractTowers();
se->registerSubsystem( st );

JetReco *towerjetreco = new JetReco();
towerjetreco->add_input(new TowerJetInput(Jet::CEMC_TOWER_SUB1));
towerjetreco->add_input(new TowerJetInput(Jet::HCALIN_TOWER_SUB1));
towerjetreco->add_input(new TowerJetInput(Jet::HCALOUT_TOWER_SUB1));
towerjetreco->add_algo(new FastJetAlgoSub(Jet::ANTIKT,0.4),"AntiKt_Tower_r04_Sub1");
towerjetreco->set_algo_node("ANTIKT");
towerjetreco->set_input_node("TOWER");
se->registerSubsystem(towerjetreco);
```

- In near future, will commit a “default” macro

Relevant node tree items

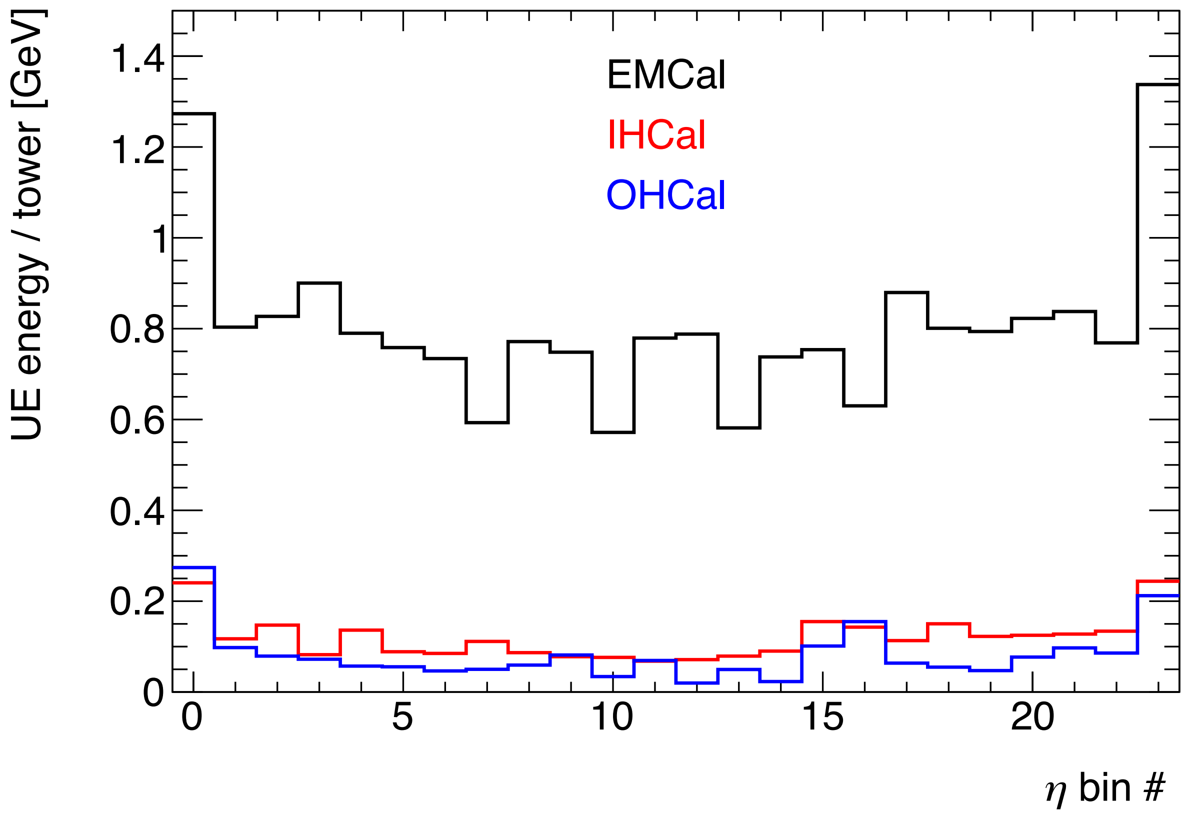
- TOP (PHCompositeNode)/
  - DST (PHCompositeNode)/
    - HCALIN (PHCompositeNode)/
      - TOWER_CALIB_HCALIN (IO,RawTowerContainer)
      - **TOWER_CALIB_HCALIN_SUB1** (IO,RawTowerContainer)
    - HCALOUT (PHCompositeNode)/
      - TOWER_CALIB_HCALOUT (IO,RawTowerContainer)
      - **TOWER_CALIB_HCALOUT_SUB1** (IO,RawTowerContainer)
    - CEMC (PHCompositeNode)/
      - TOWER_CALIB_CEMC (IO,RawTowerContainer)
      - **TOWER_CALIB_CEMC_RETOWER** (IO,RawTowerContainer)
      - **TOWER_CALIB_CEMC_RETOWER_SUB1** (IO,RawTowerContainer)
    - ANTIKT (PHCompositeNode)/
      - TOWER (PHCompositeNode)/
        - AntiKt_Tower_r04 (IO,JetMapV1)
        - **AntiKt_Tower_r04_Sub1** (IO,JetMapV1)
    - JETBACKGROUND (PHCompositeNode)/
      - **TowerBackground_Sub1** (IO,TowerBackground_v1)

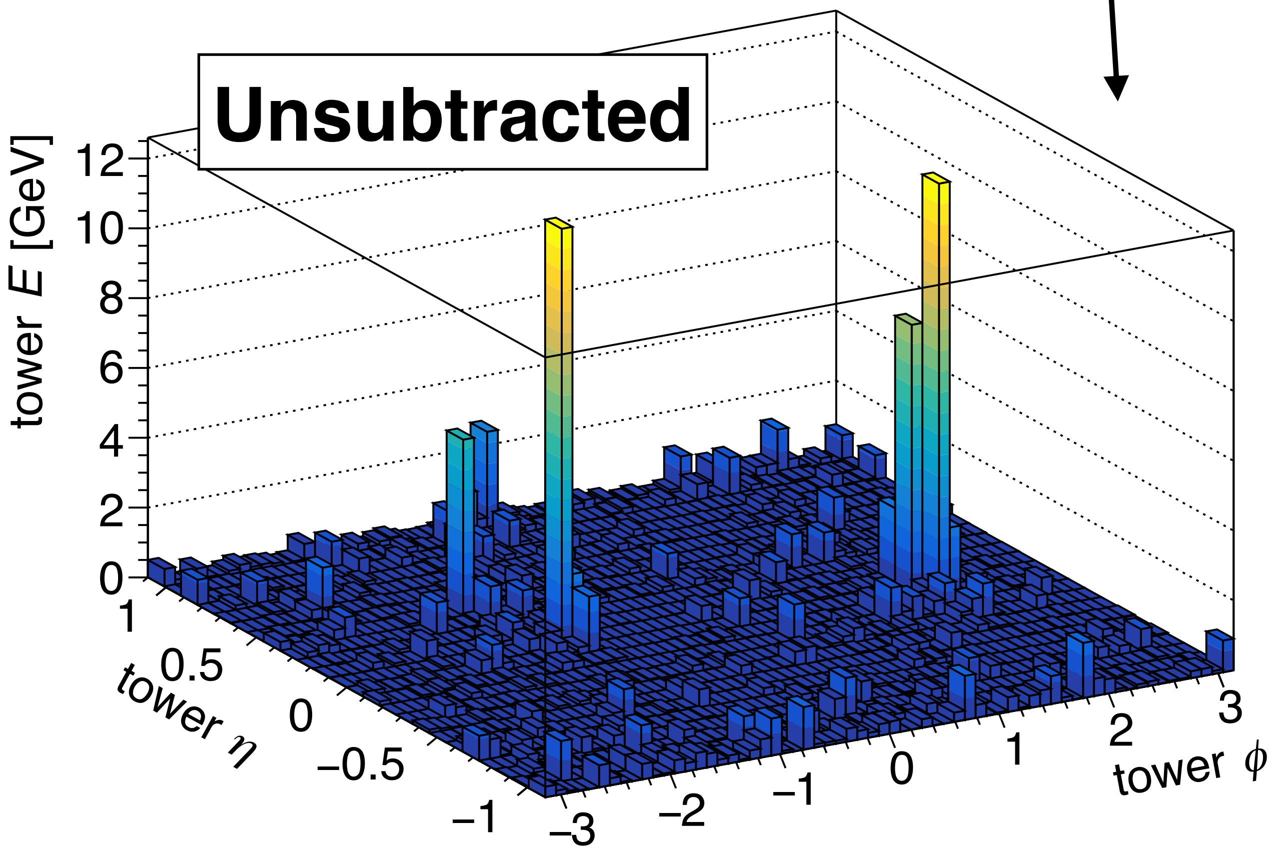
Example event

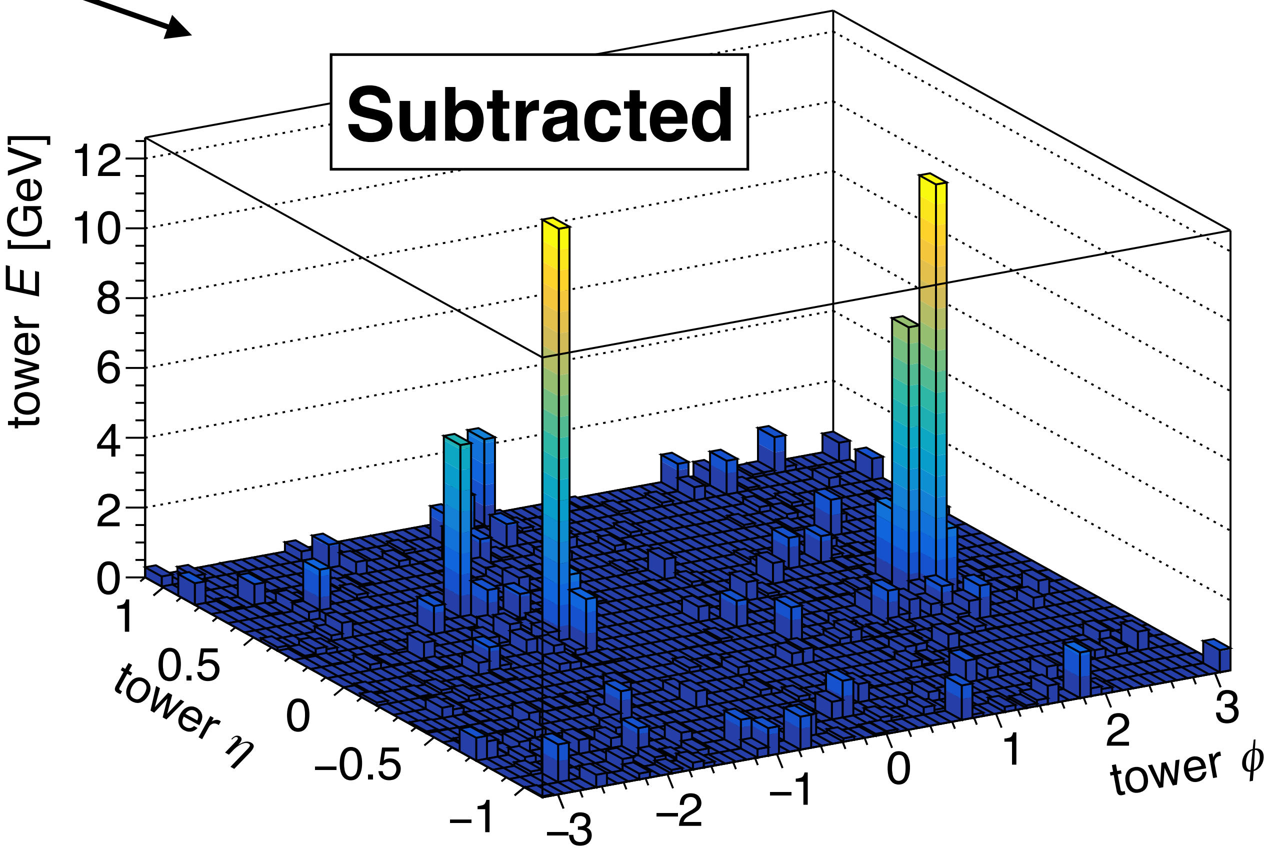
- As first “proof of concept”, analyzing procedure on a single dijet + UE background event
- QCD:Hard Pythia8 event with a generator-level filter for a R=0.4 p_{T} = 60 GeV truth jet in the middle of the sPHENIX acceptance
  - /direct/phenix+upgrades/decadal/dvp/GeneratorInputFiles/hepmc_pTHatMin50QCDall_truthtrigR0p4pT60t65eta0p6_00000_01000.dat
  - e.g. two R=0.4 truth jets ~60 GeV each, both at mid-rapidity, so should expect two R=0.4 ~45 GeV reco jets given typical response
- Embed into a b = 0-4fm sHijing hits file
  - /sphenix/data/data02/review_2017-08-02/sHijing/fm_0-4/G4Hits_sHijing_0-4fm_00000_00000.root
  - run background determination, subtraction & jet finding on subtracted towers

Background subtraction

- Estimated E / tower in different η layers
  - ➡ total ~1 GeV per tower is actually a big deal ⇒ ~50 GeV under R=0.4 jet
- η/ϕ tower energy grid

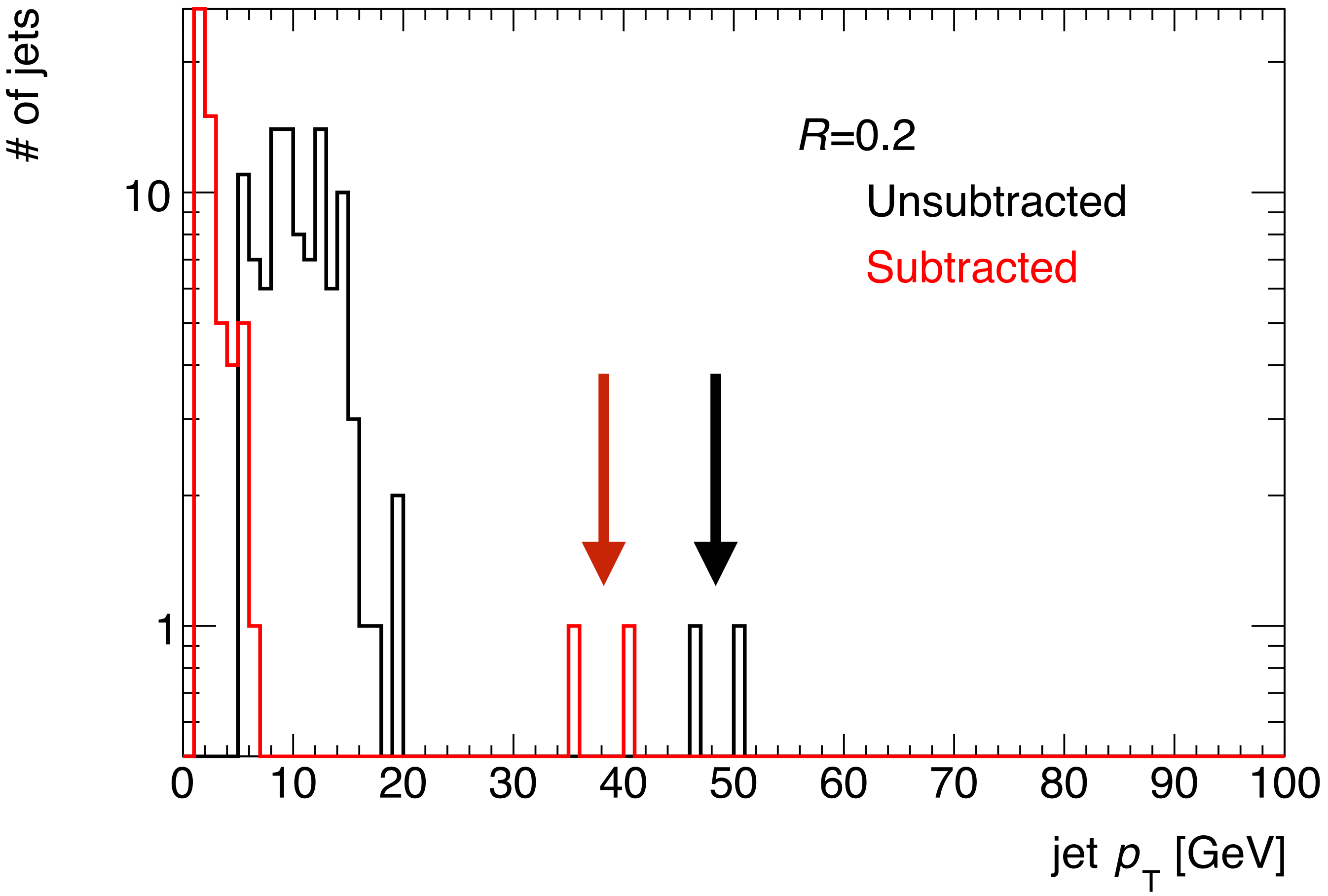


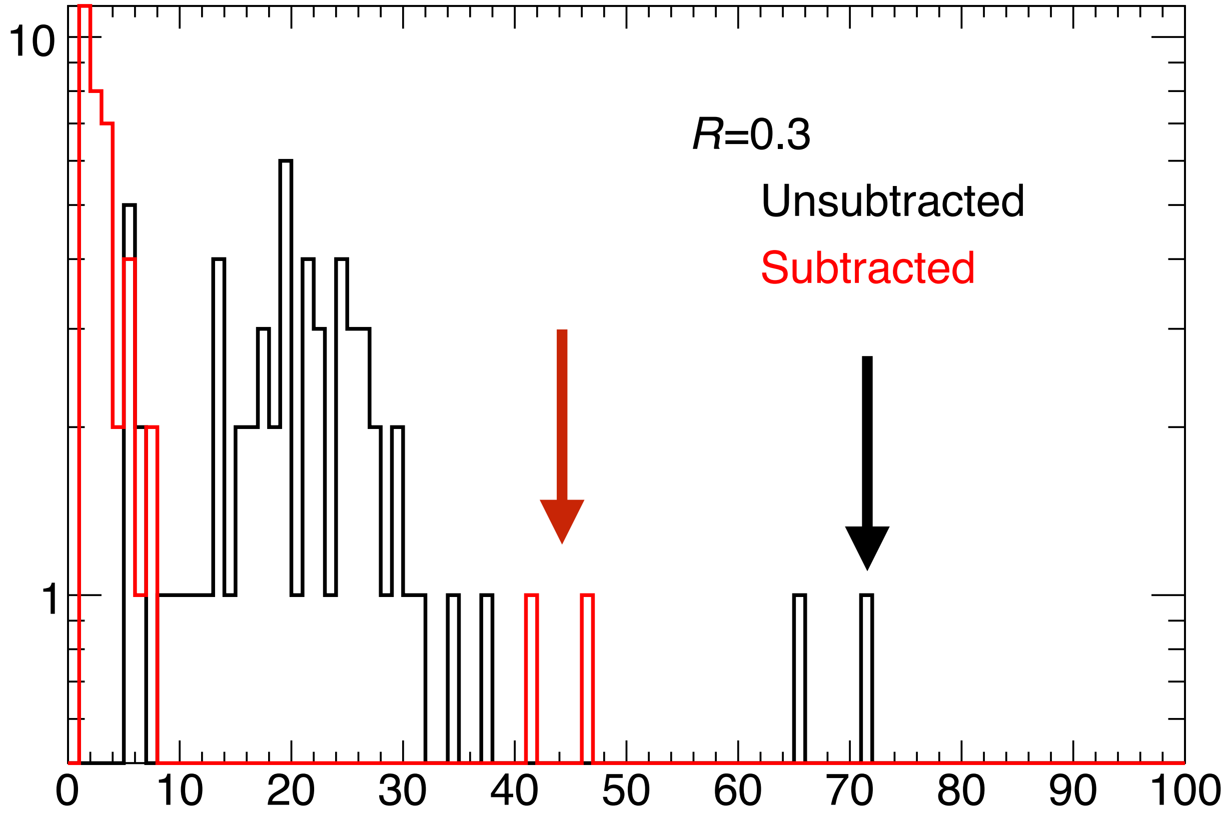


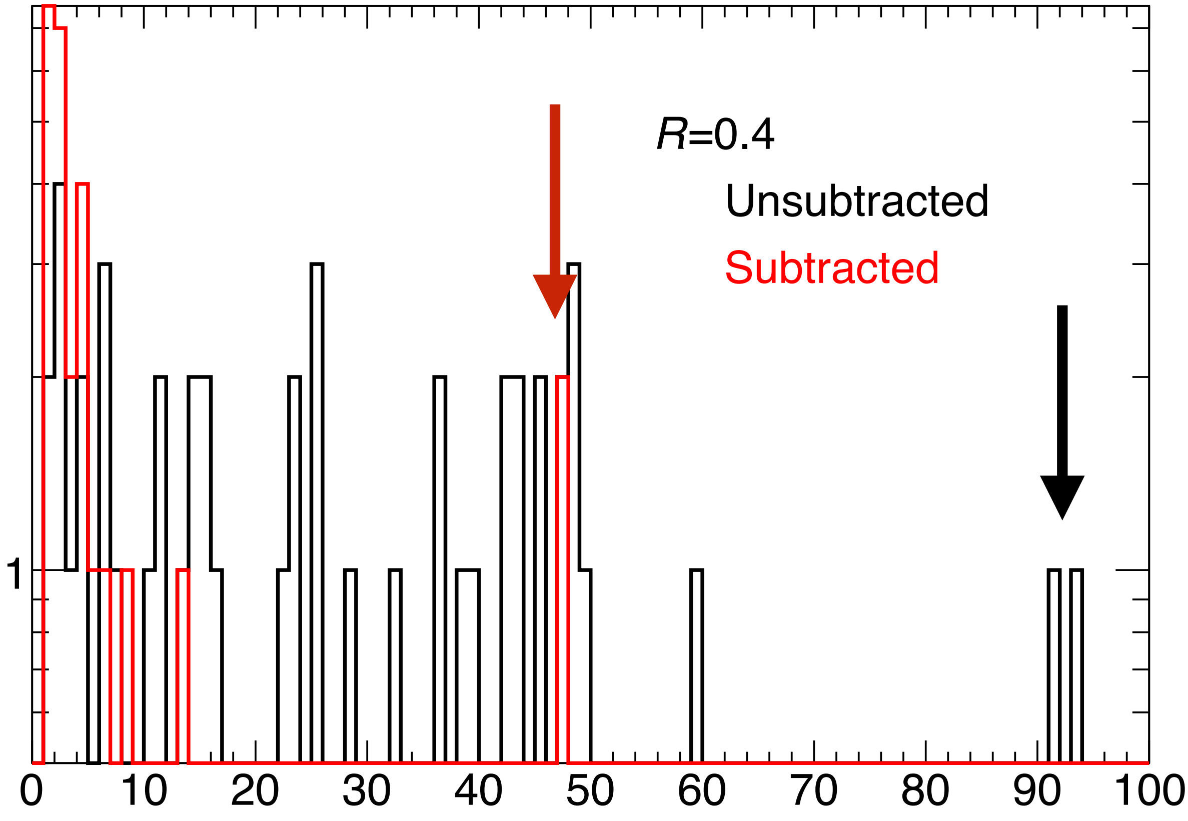


Reconstructed jet spectra

- p_T spectrum for each cone size
  - ➡ arrows point at truth-matched jets before and after subtraction
- Note that subtraction moves UE jets to much smaller p_T values







Next steps

- Additional features to be implemented:
    - ➡ v_2 / Ψ_2 determination & modulation
    - ➡ truly two-iteration procedure
    - ➡ default macro for analyzers & Wiki documentation
- Systematic study of jet performance for R=0.2, 0.3, 0.4, different sHijing b ranges, etc.
    - ➡ believe we would not need dedicated simulation production for this
        - ➡ e.g. it is time-efficient for analyzers to just embed Pythia8 events into existing G4 hits files
    - ➡ have student at Colorado, Erin Bossard, starting to look at this